TPB4 - Méthode de Monte-Carlo - Exercices

Dans tout le TP, on importe les modules suivants :

```
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt
```

Exercice 1 - Loi d'une somme, d'un produit ⋆⋆

1. On pose $X \hookrightarrow \mathcal{U}(\llbracket 1,6 \rrbracket)$ et $Y \hookrightarrow \mathcal{U}(\llbracket 1,6 \rrbracket)$ que l'on suppose indépendantes.
   (a) Quelle situation réelle est modélisée par les deux variables précédentes ?
   (b) On s'intéresse à la variable $Z = X + Y$. Préciser $Z(\Omega)$ et donner la loi de Z. Commenter alors ce que fait le code suivant :

```
vals = np.arange(2,13)
probas = np.zeros(len(vals))
for i in range(0,7):
  probas[i] = (1+i)/36
for i in range(7,12):
  probas[i] = (12-i)/36
```

   (c) Compléter le code suivant pour qu'il renvoie un tableau contenant n résultats de la somme de 2 dés simulés :

```
def mc_somme_des(n):
  sommes = np.zeros(n)
  for i in range(n):
    x = rd.randint(...) + ...
    y = rd.randint(...) + ...
    sommes[i] = ...
  return ...
```

   (d) On utilise le code suivant pour estimer la probabilité de chaque résultat par la méthode de Monte-Carlo :

```
def mc_estim_probas(vals,n):
  sommes = mc_sommes_des(n)
  probas_simul = np.zeros(len(vals))
  for i in range(len(vals)):
    filtre = (sommes == vals[i])
    nb_simul = len(sommes[filtre])
    probas_simul[i] = nb_simul/len(sommes)
  retirn probas_simul
```

   Expliquer le code précédent puis l'utiliser pour estimer la loi de Z avec 1000 simulations de lancers.
   (e) Afficher la courbe de `probas` et celle de `probas_simul` de manière superposée. Compléter le code pour afficher plutôt les fonctions de répartition réelle et estimée.
2. Adapter le code précédent pour afficher la loi de $W = X \times Y$.

Exercice 2 - D'après EDHEC 2017 ⋆⋆

Pour $k \in \mathbb{N}^\star$, on souhaite déterminer une valeur approchée de $I_k = \int_1^{+\infty} t^k e^{-t} dt$.

1. Justifier la convergence de I_k puis avec un changement de variable, montrer $I_k = e^{-1} \displaystyle\int_0^{+\infty} (x+1)^k e^{-x} dx$.
2. Soit $X \hookrightarrow \mathcal{E}(1)$. Montrer alors que pour tout $k \in \mathbb{N}^\star$, on a $I_k = \frac{E((X+1)^k)}{e}$.

3. Écrire un code permettant d'estimer I_k.

Exercice 3 - Estimation de l'erreur ⋆⋆⋆

Revenons à la simulation de Monte-Carlo pour calculer π : `mc_pi` du TP précédent. On va chercher à représenter l'ordre de grandeur de l'erreur de l'estimation.

1. Pour cette première question, on va fixer le paramètre `n` de `mc_pi` à 100.
   (a) Faire 1000 estimations de π avec autant d'appels à `mc_pi` et stocker les résultats dans un tableau `resultats`.
   (b) Utiliser le code suivant :

```
plt.hist(resultats,15)
plt.show()
```

   Que fait ce code ? Commenter le résultat obtenu.
   (c) On écrit la fonction suivante :

```
def mc_estim_var(n):
  resultats = ...
  for i in range(1000):
    resultats[i] = mc_pi(...)
  return np.var(resultats)
```

   Compléter la fonction comme précédemment. Que fait cette fonction ?
2. On part du code suivant :

```
tab_n = np.arange(50,1001,50)
variances = np.zeros(len(tab_n))
for i in range(len(tab_n)):
  variances[i] = mc_estim_var(tab_n[i])
```

   (a) Que fait ce code ?
   (b) Utiliser le code suivant :

```
plt.plot(tab_n,variances)
plt.show()
```

   Que fait ce code ? Commenter la courbe obtenue.
   (c) Soit $f : \mathbb{R}_+^\star \to \mathbb{R}_+^\star, x \mapsto f(x)$. Montrer que $\ln \circ f$ est une fonction affine de $\ln(x)$ si et seulement si f est proportionnelle à une puissance de x.
   (d) Utiliser le code suivant :

```
plt.plot(np.log(tab_n),np.log(variances))
plt.show()
```

   Constater que la courbe est *approximativement* une droite. En estimer le coefficient directeur. Commenter.

Exercice 4 ⋆⋆⋆

On pose $X \hookrightarrow \mathcal{U}([0,1])$ et $Y \hookrightarrow \mathcal{U}([0,1])$ indépendantes. On pose $Z = X + Y$.

1. Écrire une fonction qui simule n tirages de Z.
2. En s'inspirant de l'exerice **??**, écrire une fonction `mc_somme_uniforme` qui prend un tableau de valeurs de x, ainsi qu'un nombre n de tirages, et renvoie les estimations de $F_Z(x)$.
3. Tracer la courbe de la fonction $F_Z(x)$ obtenue pour $n = 1000$. Quelle forme a la courbe ?